

## Übungsblatt 02

Ausgabe: 04.11. Abgabeschluss: Mittwoch, 9.11., 9:45 Uhr, [eClaus.informatik.uni-stuttgart.de](http://eClaus.informatik.uni-stuttgart.de)

---

Ablauf der Übungen: Ausgabe jeweils in der Übung, Abgabe jeweils 9:45 Uhr am Tag der folgenden Übung (also 9.11., 16.11., jeweils 9:45 Uhr). Abgabe erfolgt ausschließlich elektronisch über [eClaus.informatik.uni-stuttgart.de](http://eClaus.informatik.uni-stuttgart.de) – versuchen Sie nach Möglichkeit die Abgabe nicht in der letzten Minute zu machen!

Von jedem Aufgabenblatt werden maximal 20 Punkte auf den Schein angerechnet.

---

1. (2+2 Punkte) **Programmanalyse:** Das AdaLogo-Programm

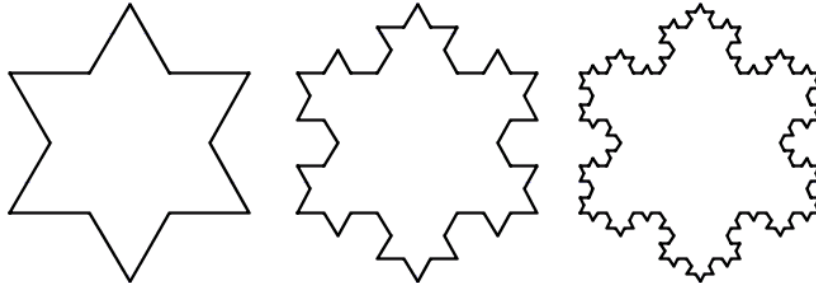
```
-----  
-- Autor: sl - Übungsblatt 02 - Aufgabe 1  
-- Datum 04.11.2005  
-- Frage: Welches Ergebnis wird in Abhängigkeit  
--       des Wertes von zahl ausgegeben?  
-- Idee : ?  
-----  
  
with adalogo;  
use adalogo;  
  
procedure Was_Tut_Das is  
  zahl: integer := 42;  
  erg : integer := 0;  
begin  
  while erg*erg <= zahl loop  
    erg := erg + 1;  
  end loop;  
  erg := erg - 1;  
  Put("Das Ergebnis lautet: "); Put(erg); New_Line;  
end;
```

berechnet eine Funktion  $f : \mathbb{Z} \rightarrow \mathbb{Z}$  (Inhalt der Variable `erg` am Ende des Programms in Abhängigkeit des Inhalts der Variable `zahl` zu Beginn).

- (a) (2 Punkte) Führen Sie das Programm von Hand schrittweise für verschiedene Anfangswerte der Variablen `zahl` durch. Welche Funktion  $f$  wird durch das Programm `Was_Tut_Das` realisiert? Begründen Sie Ihre Antwort.
  - (b) (2 Punkte) Wieviel Schritte benötigt das Programm zur Berechnung des Ergebnisses (abhängig von der Anfangsbelegung der Variablen `zahl`)? Begründen Sie Ihre Antwort.
  - (c) **Zusatzaufgabe (schwer, 4 Punkte):** Entwerfen Sie einen Algorithmus, der das Ergebnis mit deutlich weniger Schritten berechnet. Programmieren Sie diesen Algorithmus in AdaLogo aus. Fügen Sie dabei die Idee als Kommentarzeilen mit in das Programm ein und schätzen Sie ab, wieviel Schritte Sie nun zur Berechnung benötigen.
2. (5+5 Punkte) **Zeichnen selbstähnlicher Figuren:** Fraktale sind schöne Beispiele für Rekursion in der Programmierung.
- (a) (5 Punkte) Koch'sche Schneeflockenkurve (KSK): Man beginnt mit einem Dreieck (KSK der Stufe 0). Um aus dem Dreieck die KSK der Stufe 1 zu erhalten, ersetzt man bei jeder Kante das Mittelstück durch eine dreiecksähnliche Ausbuchtung, das Ergebnis ist der links abgebildete Stern. Ersetzt man bei diesem wiederum bei jeder Kante das Mittelstück

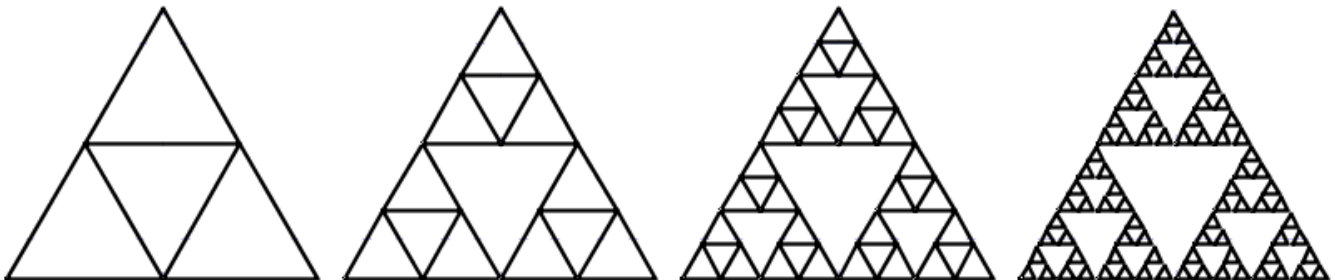
durch zwei Kanten derselben Länge, so erhält man die KSK der Stufe 2 (in der Abbildung in der Mitte). Die KSK der Stufe 3 ist rechts abgebildet.

Schreiben Sie eine Prozedur Schneeflocke, die als Parameter die Kantenlänge und die Stufe übergeben bekommt und die entsprechenden Koch'sche Schneeflockenkurve zeichnet. Dabei werden natürlich nicht schon gezeichnete Kanten gelöscht und neu gezeichnet, sondern Sie müssen sich überlegen, wie die KSK gleich richtig gezeichnet wird. (Beachten Sie, dass beim Zeichnen immer wenigstens 1 Schritt gezeichnet wird – testen Sie Ihr Programm mit den Eingaben (2,100), (3,100), (3,8) und anderen Ihrer Wahl)



- (b) (5 Punkte) Sierpinski-Dreieck (SD): Auch hier ist der Ausgangspunkt ein Dreieck (Stufe 0). Das SD der Stufe 1 erhält man, indem man statt dem großen Dreieck, drei kleinere Dreiecke mit jeweils halber Kantenlänge zeichnet. Analog erhält man das SD der jeweils nächsten Stufe, indem statt einem Dreieck jeweils drei kleinere Dreiecke mit jeweils halber Kantenlänge gezeichnet werden. In der Abbildung sind von links nach rechts die Sierpinski-Dreiecke der Stufen 1, 2, 3 und 4 abgebildet.

Schreiben Sie eine Prozedur Sierpinski, die als Parameter die Kantenlänge und die Stufe übergeben bekommt und das entsprechende Sierpinski-Dreieck zeichnet. Es gilt der gleiche Hinweis wie bei der ersten Teilaufgabe.



3. (3 Punkte) **Freitag der 13.:** Ist der Wochentag des 1. Januar bekannt, so lässt sich eindeutig bestimmen, auf welchen Wochentag ein Datum dieses Jahres fällt (wir gehen von Jahren aus, die kein Schaltjahr sind, also 28 Tage im Februar).

Schreiben Sie eine Prozedur, die in Abhängigkeit des Wochentags am 1. Januar bestimmt, in welchen Monaten der 13. auf einen Freitag fällt.

4. (2+3 Punkte) **Umsetzung in Ada 95:** Die Übungen werden in Kürze hauptsächlich in Ada 95 durchgeführt. Diese Aufgabe soll Sie langsam dazu hinführen.

- (a) (2 Punkte) Schreiben Sie das Programm aus Aufgabe 1 in Ada 95 um und kommentieren Sie es so, dass es gut verständlich ist (Sie dürfen dazu auch Variablennamen ändern).

- (b) (3 Punkte) Fügen Sie einige (mindestens 6) Syntaxfehler in Ihr Programm ein. Beschreiben Sie, welche 6 Fehler Sie eingefügt haben, geben Sie die Fehlermeldung des Compilers an und beurteilen Sie, ob diese Fehlermeldung hilfreich ist und den Fehler aus Ihrer Sicht zutreffend beschreibt. Die 6 Fehler sollten folgende Fälle beinhalten: zusätzliches Semikolon, fehlendes Semikolon, fehlendes Trennzeichen, Tippfehler bei Variablennamen, „=“ statt „:=“ bei einer Zuweisung.